**CS223**

**DIGITAL DESIGN**

**LAB 03**

**BERK YILDIZ**

**21502040**

**PACK NUMBER 31**

**SystemVerilog module for a 2-bit up-counter**

module 2bitupcounter(clk, in, preset, reset, count);

input[1:0] data\_in;

input clk, reset, preset;

output reg [1:0]count ;

always @(posedge clk)

begin

If(reset==1)

count <=0;

else if(preset==1)

count<=in;

else

count <=count+1;

end

endmodule

**SystemVerilog module for a 3-bit gray code counter**

module 3-bitGrayCodeCounter(input logic clk\_in,

                input logic preset,

                input logic reset,

                output logic[2:0] out);

                logic clk\_out;

         ClockDivider clock(clk\_in, clk\_out);

        always\_ff @(posedge clk\_in)

        begin

        if (reset)

        out <= 0;

       else if (preset)

        out <= 3'b111;

        else

        case(out)

        3'b000:  out <= 3'b001;

        3'b001:  out <= 3'b011;

        3'b011:  out <= 3'b010;

        3'b010:  out <= 3'b110;

        3'b110:  out <= 3'b111;

        3'b111:  out <= 3'b101;

        3'b101:  out <= 3'b100;

        3'b100:  out <= 3'b000;

        default: out <= 3'b000;

        endcase

        end

endmodule

**Testbench for 3-bit gray code counter**

module testGrayCodeCounter();

    logic clk\_in, preset, reset;

    logic [2:0] out;

   3-bitGrayCodeCounter dut(clk\_in, preset, reset, out);

   always begin

     clk\_in = 0; #30; clk\_in = 1; #30;

   end

   initial begin

    reset = 0; preset=0; #300;

    reset = 1; #100;

    reset = 0; preset=1; #100;

   end

endmodule